



IMT Mines Albi-Carmaux
École Mines-Télécom

Projet Système d'Information

Rendu phase 3

Rapport sur la Gestion de Projet Tutor Link

Groupe OEM

Muneeb MIRZA - Julien LAGARDE - Ghali SLAOUI - Meanonn TAN

1. Gestion de projet	3
• Répartition des tâches	3
• Avancement du projet	3
2. Les problèmes rencontrés	4
• Problème de codage	4
• Problème de gestion de base de données	5
• Si c'était à refaire...	5
3. Les suites à donner au produit	6
• Les features manquantes	6

1. Gestion de projet

● Répartition des tâches

Pour réaliser le projet efficacement et pour un bon travail d'équipe, nous nous sommes réparti les tâches techniques et managériales. En effet, étant un groupe de 4, nous avons dans un premier temps décidé de séparer le management de l'équipe des tâches techniques.

La tâche managériale consistait principalement à gérer l'excel de suivi, pour voir quelles sont les tâches urgentes, lesquelles sont bientôt finies, pour, lors d'une réunion de milieu de semaine, prévenir les 3 autres qui développent.

Les tâches techniques étaient la répartition de l'écriture du code. Nous avons commencé par une répartition back-end, front-end et test dans un premier temps, puis nous sommes passés sur une répartition en 2 équipes, une avec un développeur seul et l'autre en peer-programming qui nous a permis de former en même temps que coder le développeur du peer-programming. Chaque équipe s'occupait alors de son propre blueprint (front back et test). Cela a également changé la manière de créer les branches du repo git.

Cela n'a pas changé le travail de la personne chargée de la tâche managériale.

● Avancement du projet

Pour réaliser la répartition des tâches nous avons utilisé un tableur partagé qui nous a permis d'avoir une liste des tâches manquantes et de l'avancé des tâches.

Project Start 01/01/2023						
TASK	OWNER	PRIORITY	START	END	% COMPLETE	Done
Les Vues						
CAS	Ghall	LOW	01/01/2023	25/01/2023	100%	○
Accueil	Muneeb	HIGH	01/01/2023	1/22/2023	100%	○
Mes Cours	Ghall	HIGH	01/01/2023	21/01/2023	100%	○
Panel Admin	Ghall	MEDIUM	22/01/2023		90%	○
Respo Module	Muneeb	MEDIUM	26/01/2023	15/02/2023	100%	○
Routes						
GET liste_séances	Julien	HIGH	07/01/2023	27/01/2023	100%	○
POST ajouter_intervenant	Julien	MEDIUM	07/01/2023	30/01/2023	100%	○
POST retirer_intervenant	Julien	MEDIUM	07/01/2023	29/01/2023	100%	○
GET role_list	Muneeb	HIGH	23/01/2023	05/02/2023	100%	○
POST add_role	Julien	MEDIUM	24/01/2023	06/02/2023	100%	○
POST delete_role	Julien	MEDIUM	25/01/2023		90%	○
POST add_favorite	Muneeb	LOW	25/01/2023		90%	○
POST delete_favorite	Julien	LOW	25/01/2023		90%	○
GET dashboard_module	Meanon	HIGH	28/01/2023	17/02/2023	100%	○
POST notifier	Meanon	MEDIUM	28/01/2023	17/02/2023	100%	○
POST login	Meanon	HIGH	30/01/2023	21/02/2023	100%	○

Un point entre les membres du groupe est fait en début de semaine pour la répartition des tâches, puis un autre en milieu de semaine pour la mise en commun, pour aider une personne si celle-ci est bloquée ou si elle a besoin d'aider et ainsi se fixer de nouveaux objectifs pour la fin de chaque semaine en prenant en compte les problèmes évoqués.

Ces réunions avaient le défaut avec la première répartition (front d'un côté et back de l'autre) de s'éterniser, les vitesses de développement étaient très différentes car les compétences de chacun en début de projet étaient très hétérogènes ainsi que la compréhension globale du projet et du développement d'une application avec Flask, c'est pour cela que nous avons décidé de faire du peer-programming. Cela ralentissait certains d'entre nous mais ce temps n'était pas perdu car les autres membres se formaient/comprenaient beaucoup plus vite.

2. Les problèmes rencontrés

- **Problème de codage**

Pour réaliser convenablement l'application nous devions nous familiariser avec le fonctionnement de bootstrap et de flask. Cela n'a pas été un réel problème en soi mais il nous a fallu un temps d'adaptation qui était différent pour chacun. Le peer-programming a grandement aidé mais cela reste, en termes de temps, bien au-dessus des estimations au début de cette phase-3.

Au fur et à mesure du développement nous avons procédé à des tests de l'application qui nous permet de repérer les bug potentiels. Ce point là était le point le plus sous-estimé de tous. En effet, la quantité de tests étant très conséquente et nous n'avions absolument pas anticipé cela. Cependant, nous sommes restés sur notre position, à savoir continuer à réaliser de nombreux tests unitaires, car ils nous permettent d'affirmer avec certitude que notre application est très robuste. Même si cela nous a empêché de réaliser plusieurs fonctionnalités de l'application, nous avons préféré faire moins, mais de bien meilleure qualité. Enfin, le coverage global des tests nous a confortés dans notre choix et nous a rendu bien plus fiers de notre produit.

● Problème de gestion de base de données

Au niveau de la récupération des données sur le fichier JSON de synapses, un travail a été réalisé sur l'adaptation des données (La récupération des bonnes données) . Il a fallu adapter toute l'application à la spécification de ce fichier.

Nous pouvions difficilement anticiper le temps de développement sur cette partie du projet car nous ne savions pas à quoi nous attendre (quelles données on pouvait avoir, si elles avaient besoin de beaucoup de pré-processing ou non, si on pouvait faire une maj automatique ou manuelle...).

Nous avons volontairement anticipé une grosse durée pour cette partie, mais les tests de notre automatisation d'intégration les données Synapses à notre base de données ont tout de même pris plus de temps que notre large anticipation.

● Si c'était à refaire...

Notre organisation sur la fin du projet nous a convaincu sur son efficacité. Cela nous a donc appris qu'il ne faut pas négliger les disparités de compétences au début du projet et prendre en compte ce critère dans nos anticipations dans nos futurs projets.

Nous avons également réalisé que l'anticipation du temps de développement sur les options doit être beaucoup plus large. En effet, les tests unitaires prennent beaucoup de temps à réaliser et c'est une information que l'on prendra en compte par la suite.

Nous avons décidé de complètement ignorer le côté responsable de notre application. En effet, nos scripts frontend, dû à la forte utilisation de Bootstrap par exemple génèrent un grand nombre de requêtes à chaque affichage ce qui consomme beaucoup plus que si nous téléchargeons les classes requises. L'optimisation du code a bien été prise en compte cependant, mais nous aurions pu l'optimiser encore plus ceci dit.

3. Les suites à donner au produit

- **Les features manquantes**

Lors de cette phase développement nous nous sommes focalisés sur les éléments essentiels au rôle primaire de l'application.

Voici une liste des features suivantes à ajouter par ordre de priorité :

- Finalisation de la vue respo. module pour permettre au responsable d'un module d'avoir une vue plus globale sur le remplissage des cours dans ses modules ou encore d'avoir une liste d'intervenants, enseignants ou encore doctorants qui sont qualifiés pour intervenir dans son module.
- Amélioration de "Admin Panel" pour permettre à l'administrateur de voir les responsables modules, de supprimer un utilisateur, d'en ajouter un ou encore d'aller voir le profil, ou le nombre d'heures réalisées d'un des utilisateurs. En effet, pour l'instant, le statut d'administrateur ne donne pas beaucoup plus de "pouvoir", si ce n'est que de pouvoir ajouter les nouvelles données de Synapses.
- Un bouton de notifications qui permettrait, en plus d'aider les utilisateurs, d'avoir un suivi des actions réalisées (inscription/désinscription par exemple) par les utilisateurs.